

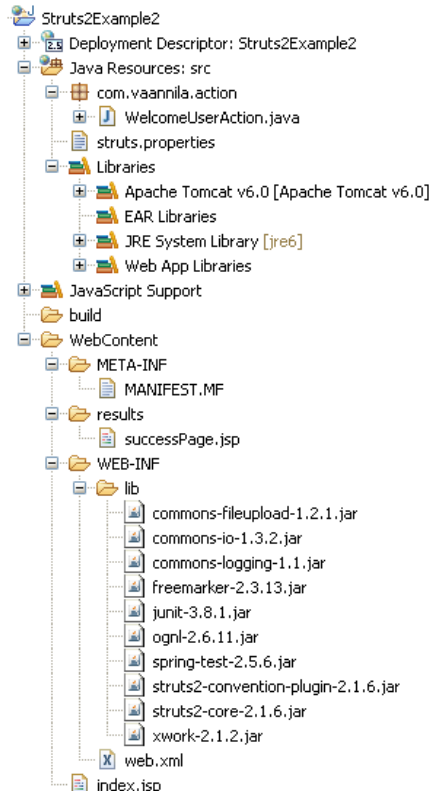
## Menu

[Struts 1](#)[Struts 2](#)[Spring](#)[Hibernate](#)[Ant](#)[Log4j](#)[Struts 2](#) > Struts 2 Annotations Example**Struts 2 Annotations Example**

This example is the continuation of the previous annotation example. If you are new to Struts 2 annotations then go through that example first ( [Struts 2 Annotations - Part 1](#) ). Here we will see the same hello user example with the following changes.

- Our Action class ends with the word **Action** and does not implement `com.opensymphony.xwork2.Action`.
- We use **/results** directory for storing our result pages instead of `WEB-INF/content`.

The directory structure of the hello user example is shown below.



Our `WelcomeUserAction` class is a simple pojo class. The important thing to note here is that our Action class name ends with the word **Action**.

```

01. package com.vaannila.action;
02.
03. import org.apache.struts2.convention.annotation.Action;
04. import org.apache.struts2.convention.annotation.Result;
05.
06. public class WelcomeUserAction {
07.     private String userName;
08.     private String message;
09.
10.
11.     @Action(value="/welcome",results={
12.         @Result(name="success",location="/results/successPage.jsp")})
13.     public String execute() {
14.         message = "Welcome " + userName + " !";
15.         return "success";
16.     }
17.
18.     public void setUserName(String userName) {
19.         this.userName = userName;
20.     }
21.
22.     public void setMessage(String message) {
23.         this.message = message;
24.     }
25.
26.     public String getUserName() {
27.         return userName;
28.     }
29.
30.     public String getMessage() {
31.         return message;
32.     }

```

Here we use Action and Result annotations just to show you how to use them, for simple example like this you can use the intelligent defaults provided by the Convention plug-in.

The Convention plug-in uses the Action class name to map the action URL. The Convention plug-in first removes the world Action at the end of the class name and then converts the camel case

## Subscribe

[RSS](#)[Email](#)

name to dashes. So by default our `WelcomeUserAction` will be invoked for the request URL `welcome-user`. But if you want the Action to be invoked for a different URL then you can do this by using the **Action annotation**.

The value of the Action annotation is `"/welcome"`, this means that the action will be invoked for the request URL `"/welcome"`. You can change the default action and URL mapping using the Action annotation.

Now based on the result code from action the Convention plug-in will look for the result name **welcome-resultcode** in the directory `WEB-INF/content`. You can change this to a different location by setting the property **struts.convention.result.path** to a new value in the Struts properties file. In this example we store the result pages in `/results` directory.

```
1. struts.properties file
2. -----
3. struts.convention.result.path=/results
```

Our result page name is `successPage.jsp`, the Convention plug-in will look for a page like `welcome.jsp` ( the file can even be a freemaker or velocity file ) since our URL is `"/welcome"`. In this case it will give an error, if we are not specifying which result it should invoke when the result is `"success"`. To do this we use the Result annotation.

The **Result annotation** maps the result code with the result page. Here the result code `"success"` is mapped to the result `"/results/successPage.jsp"`.

The annotations needs to be specified only when you are not using the default naming conventions, if you use them you can keep writing action classes and result pages without any configuration and the framework know exactly when to invoke them.

You can download the Struts 2 annotation example by clicking the download link below.

Source : [Download](#)

Source + Lib : [Download](#)